

20

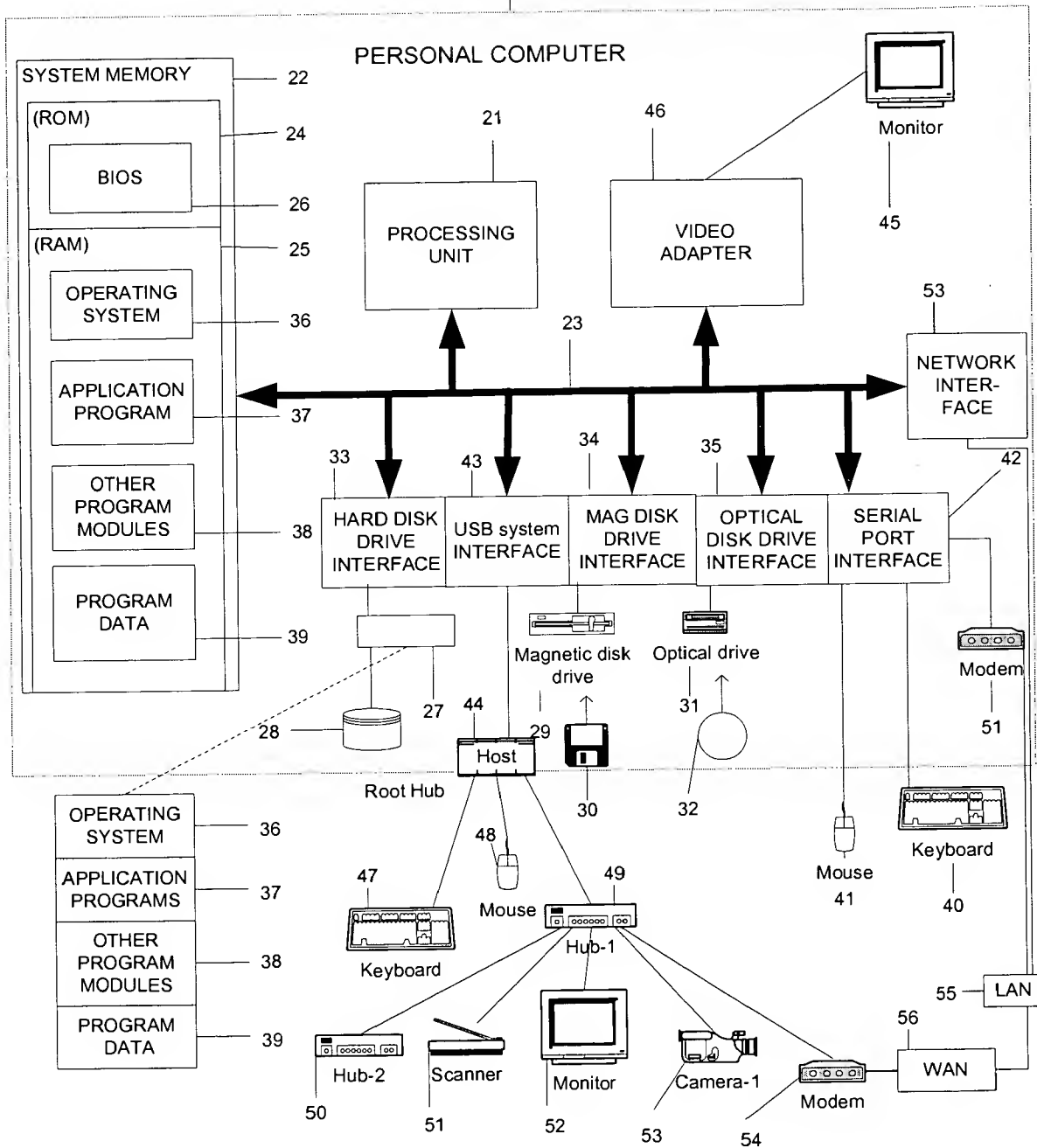


Figure 1

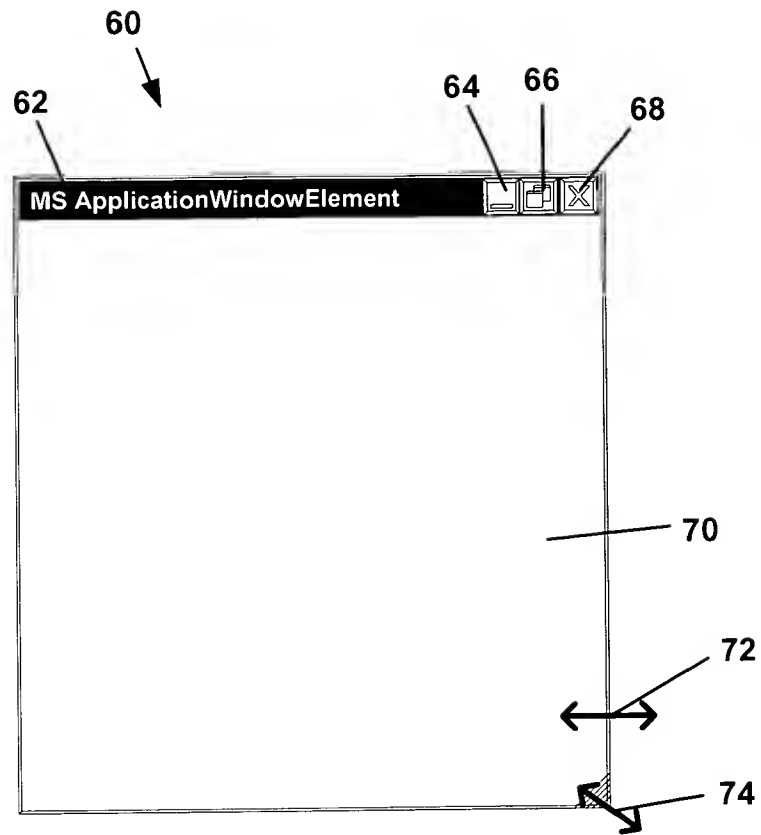
[illegible]

Figure 2

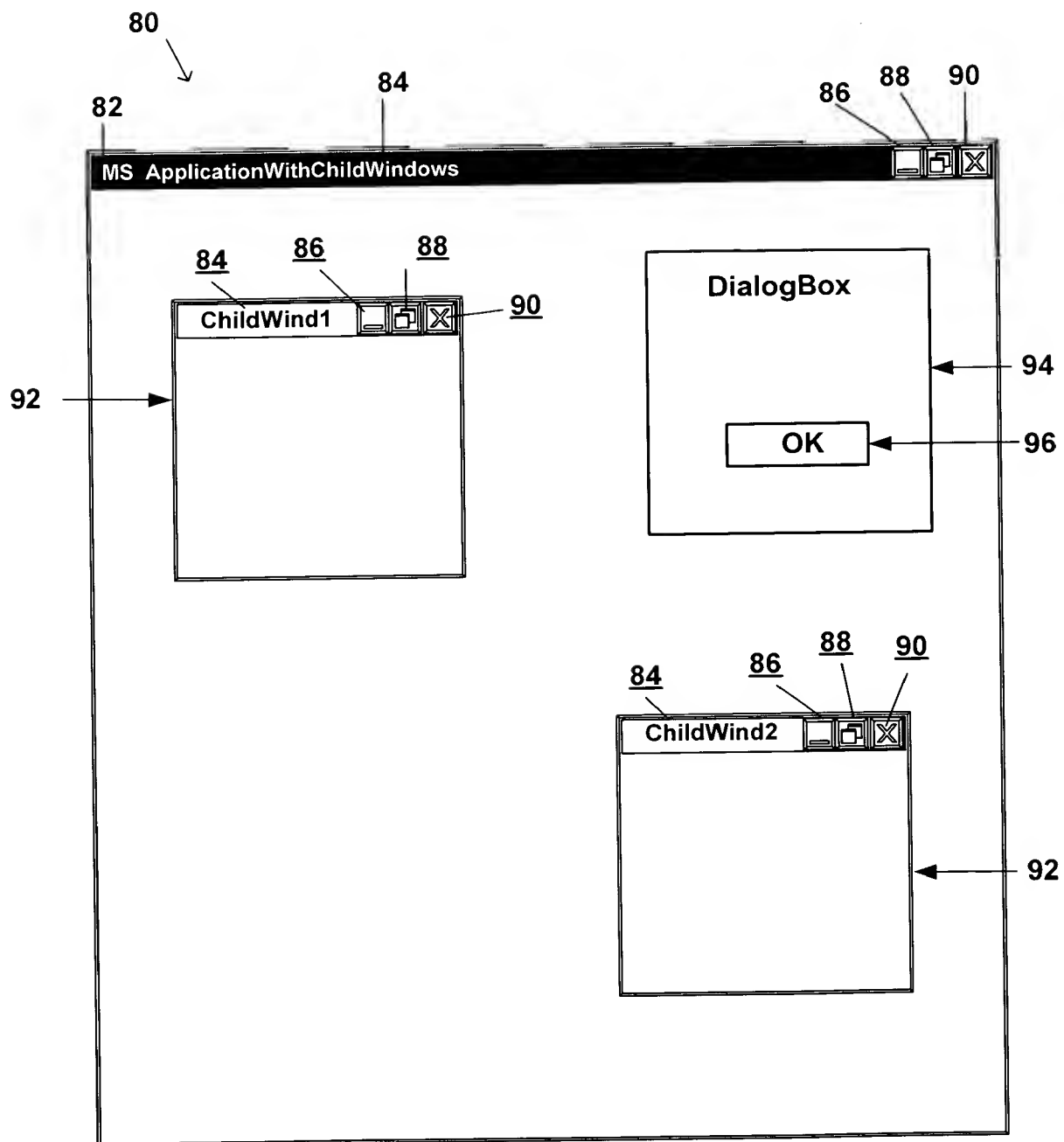


FIGURE 3

```
graph TD; 100[Desktop thread changes, in concert with the window manager, the Z-ordering of the displayed window elements] --> 102[The Window Manager identifies the applications that need to repaint due to the changes]; 102 --> 104[Desktop thread places a repaint request in the queue for each of the Application that needs to call repaint() and stamps the request with the current time]; 104 --> 106[The Relevant Application thread calls the appropriate graphical functions, if necessary, to generate the window elements to be displayed]; 106 --> 108[Application calls repaint() and sends the result to the system thread and enters current time for handling the queued event]; 108 --> 110[System receives results from Applications calling the graphical functions and repaint()]; 110 --> 112[An appropriate system thread composites the different inputs to generate the updated desktop consistent with the new Z-ordering];
```

100 Desktop thread changes, in concert with the window manager, the Z-ordering of the displayed window elements

102 The Window Manager identifies the applications that need to repaint due to the changes

104 Desktop thread places a repaint request in the queue for each of the Application that needs to call repaint() and stamps the request with the current time

106 The Relevant Application thread calls the appropriate graphical functions, if necessary, to generate the window elements to be displayed

108 Application calls repaint() and sends the result to the system thread and enters current time for handling the queued event

110 System receives results from Applications calling the graphical functions and repaint()

112 An appropriate system thread composites the different inputs to generate the updated desktop consistent with the new Z-ordering

FIGURE 4

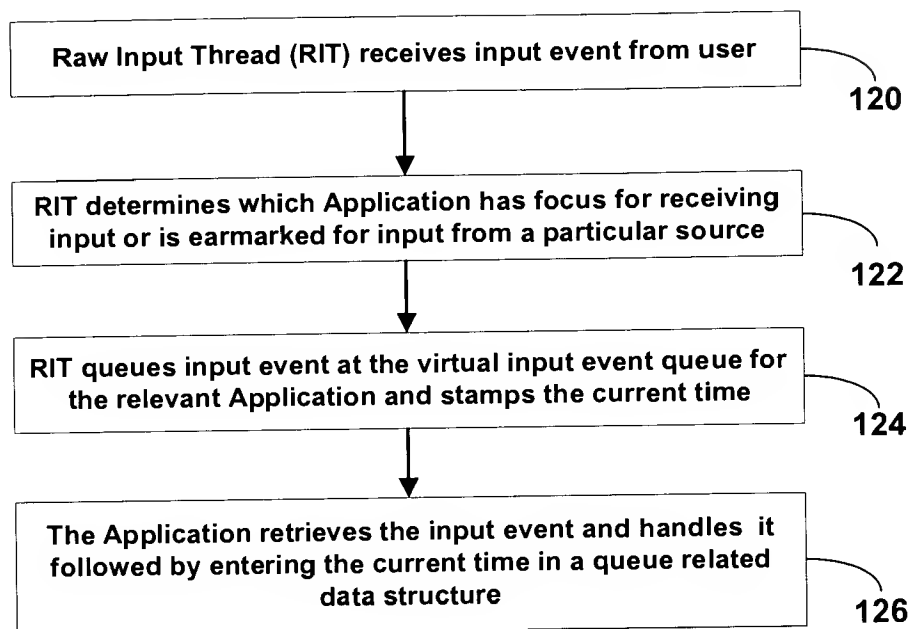
[illegible]

FIGURE 5

```

graph TD
    130[System thread detects a hung application] --> 132{System thread determines if a ghost thread exists}
    132 -- No --> 134[System thread requests that a ghost thread be created]
    132 -- Yes --> 136[System thread requests a ghost window be created by the ghost thread to cover the specified window]
    134 --> 136
    136 --> 138[Ghost thread calls graphical routines followed by repaint()]
    138 --> 140[Ghost thread updates the display system thread and focus to update the display and redirect the input]
    140 --> 142[Ghost thread is notified that the Application is no longer "hung"]
    142 --> 144[Ghost thread destroys the ghost window and restores the Application window along with queueing the input for the Application]

```

FIGURE 6

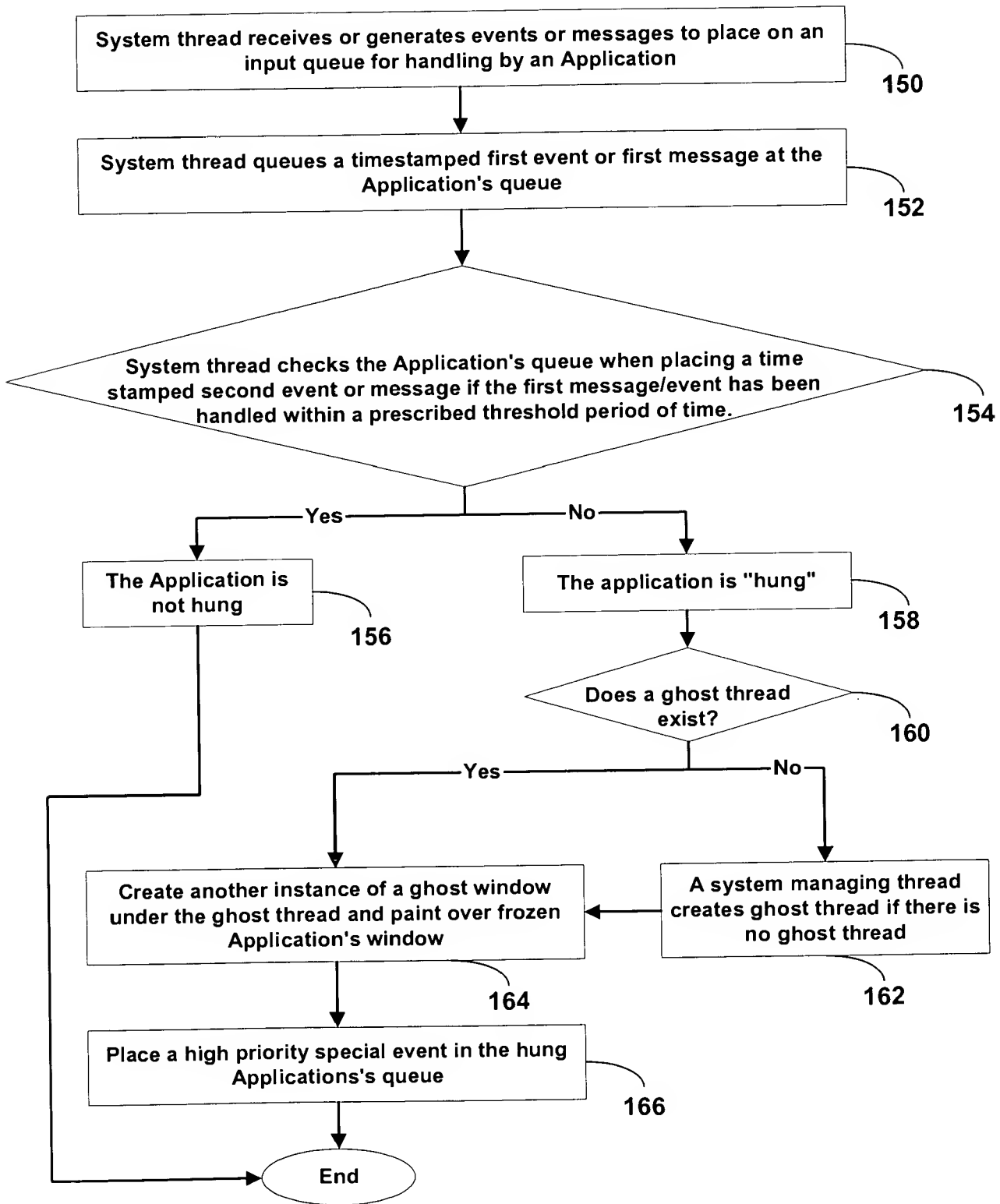


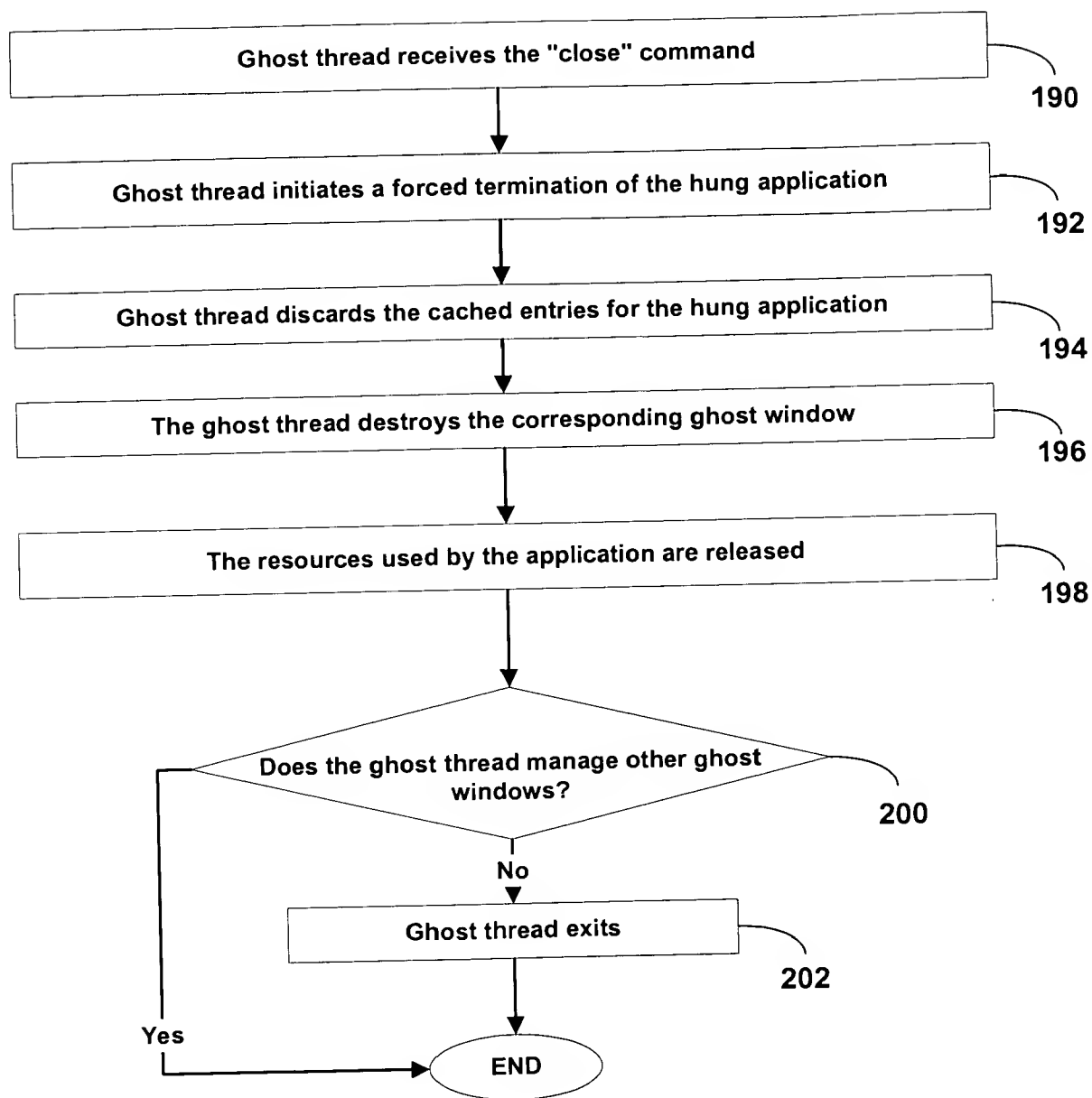
FIGURE 7

```
graph TD; 170[High priority special event moves to the front of the hung Application's queue] --> 172[Hung Application is able to handle events and messages again]; 172 --> 174[The Application handles the high priority special message by sending a message]; 174 --> 176[The ghost thread destroys the ghost window]; 176 --> 178[The ghost thread updates cached entries to remove overridden entries]; 178 --> 180[The ghost thread forwards updated cached entries to the Application's queue]; 180 --> 182{Does the ghost thread manage other ghost windows?}; 182 -- No --> 184[Ghost thread exits]; 182 -- Yes --> END([END]);
```

Flowchart 1700 illustrates the process of a ghost thread updating the cache and forwarding entries to the Application's queue. The steps are as follows:

- High priority special event moves to the front of the hung Application's queue (170).
- Hung Application is able to handle events and messages again (172).
- The Application handles the high priority special message by sending a message (174).
- The ghost thread destroys the ghost window (176).
- The ghost thread updates cached entries to remove overridden entries (178).
- The ghost thread forwards updated cached entries to the Application's queue (180).
- Decision: Does the ghost thread manage other ghost windows? (182).
 - If No, the ghost thread exits (184).
 - If Yes, the process ends (END).

FIGURE 8

**FIGURE 9**

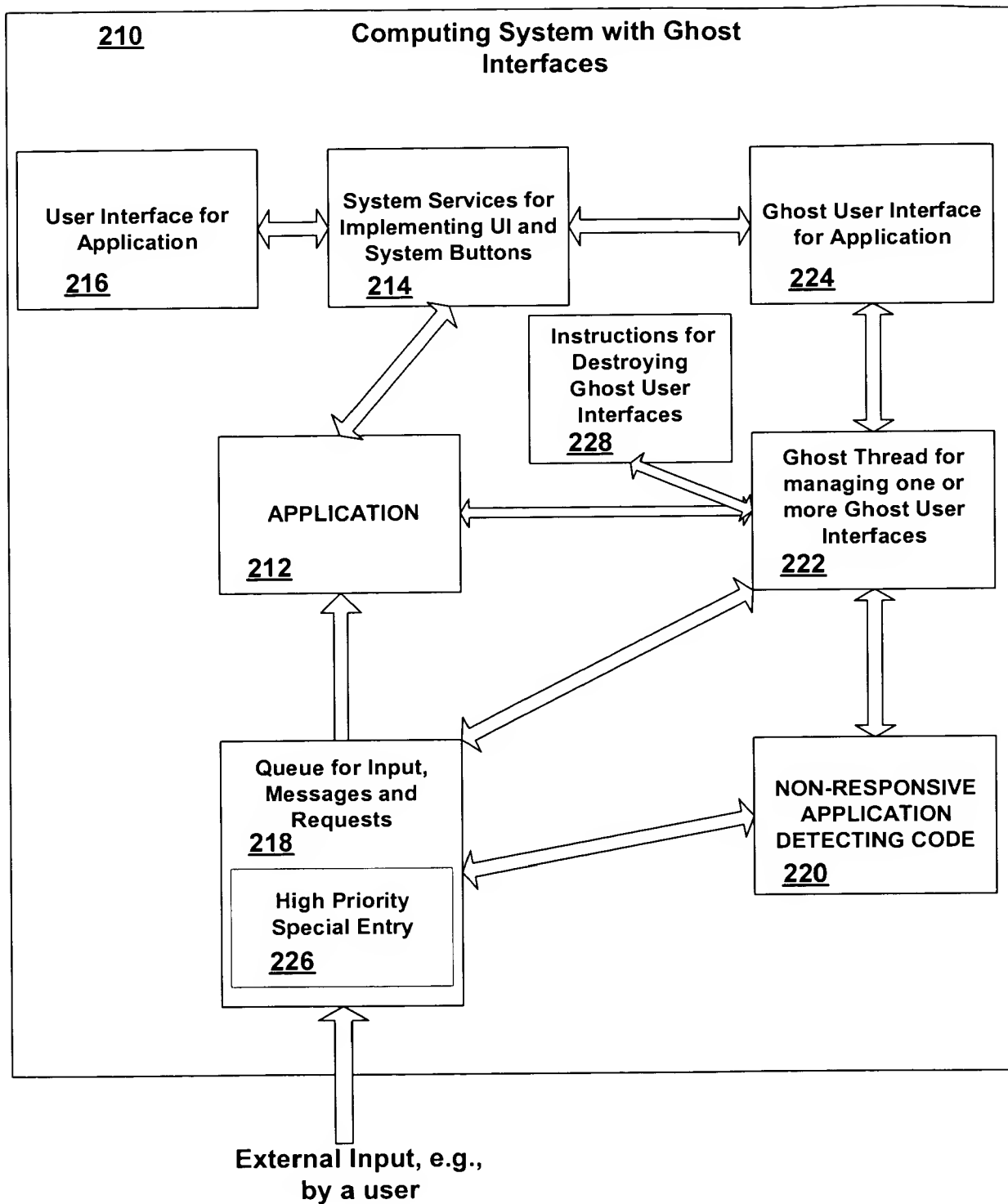


FIGURE 10